PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of   Docket No: Q61879

Fabrice BANCTEL, et al.

Appln. No.: 09/701,653   Group Art Unit: 2154

Confirmation No.: 5626   Examiner: Haresh N. Patel

Filed: November 30, 2000

For: A METHOD OF IMPLEMENTING A TREE OF DISTRIBUTED OBJECTS IN
        DIFFERENT PROCESSES USING A DATA STRUCTURE AT THE ROOT LEVEL OF
        THE TREE

SUBMISSION OF APPEAL BRIEF

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

        Submitted herewith please find an Appeal Brief. A check for the statutory fee of $500.00
is attached. The USPTO is directed and authorized to charge all required fees, except for the
Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any
overpayments to said Deposit Account. A duplicate copy of this paper is attached.

                                           Respectfully submitted,

                                           Diallo T. Crenshaw
                                           Registration No. 52,778

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: January 3, 2005

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of	Docket No: Q61879
Fabrice BANCTEL, et al.	
Appln. No.: 09/701,653	Group Art Unit: 2154
Confirmation No.: 5626	Examiner: Haresh N. Patel
Filed: November 30, 2000	

For: A METHOD OF IMPLEMENTING A TREE OF DISTRIBUTED OBJECTS IN DIFFERENT PROCESSES USING A DATA STRUCTURE AT THE ROOT LEVEL OF THE TREE

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.37, Appellant submits the following:

Table of Contents

I. REAL PARTY IN INTEREST

Based on the information supplied by the Appellants, and to the best of Appellants' legal

representative's knowledge, the real party in interest is the assignee, ALCATEL. The

Assignment was recorded on November 30, 2000 at Reel 11416, Frame 0756.

II.   RELATED APPEALS AND INTERFERENCES

Appellants, as well as Appellants' assigns and legal representatives, are unaware of any appeals or interferences which will be directly affected by, or which will directly affect or have a bearing on, the Board's decision in the pending case.

III.   STATUS OF CLAIMS·

Claims 1-9 are pending in the application, have been finally rejected, and are the subject

of this appeal. Claims 1-9, as finally rejected and appealed, are set forth in the Appendix.

Claims 1-9 (rejected).

Claims 1-8 stand rejected under 35 U.S.C.§ 102(e) as allegedly being anticipated by Rich

et al. (U.S. Patent No.: 6,457,065). Claim 9 stands rejected under 35 U.S.C. § 103(a) as

allegedly being unpatentable over Rich in view of Applicants' admitted prior art (AAPA).

IV.   STATUS OF AMENDMENTS

No amendments have been filed subsequent to the final office action.

V.   SUMMARY OF THE CLAIMED SUBJECT MATTER

In an exemplary embodiment, the present invention provides a method of implementing a tree of distributed objects in different processes (Fig. 2), wherein a central directory is adapted to store information on objects in a data structure Tab0 at the root of the tree, the method comprising assigning to a father object A in a process Pr0, for each son object B, C: information corresponding to a physical address if a son object is contained in same process, or information referring back to said central directory if the son object is not contained in the same process (*see page 4, lines 2-7, and claim 1*).

Another aspect of the above-discussed method provides, wherein if the central directory receives a request for access to a first object identified by a logical name identifying a logical access path of said first object from the central directory, the central directory searches its data structure for the logical name received in order to send the request directly to said first object or, if said logical name is not in the central directory, the central directory searches for a logical name with the longest character string equal to a first part of the character string of the logical name received, in order to send, to a father object, the request relating to the first object, by providing said father object with information corresponding to the logical access path of the first object relative to the father object (*see page 5, lines 21-30, and claim 2*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-8 stand rejected under 35 U.S.C.§ 102(e) as allegedly being anticipated by

Rich et al. (U.S. Patent No.: 6,457,065).

2. Claim 9 stands rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over

Rich in view of Applicants' admitted prior art (AAPA).

VII.   ARGUMENT

A. Claims 1-8 are NOT anticipated by Rich.

> A.1.   <u>Rich does not teach or suggest at least, "assigning to a father object in a process, for each son object, ... information referring back to said central directory if the son is not contained in the same process," as described by independent claim 1 and the claims that depend therefrom.</u>

To anticipate a claimed invention, a prior art reference must teach or suggest each and every limitation of the claimed invention. Here, Appellants submit that Rich does not teach or suggest each and every limitation of independent claim 1.

The present invention, as recited in claim 1, is directed to a method of implementing a tree of distributed objects and different processes, wherein there is a central directory adapted to store information on objects in a data structure at the root of the tree; the method comprises assigning to a father object in a process, for each son object: information corresponding to a physical address if a son object is contained in a same process, or information referring back to the central directory if the son object is not contained in the same process.

Rich, on the other hand, relates to a technique for replicating objects in distributed object systems to reduce network round trips during program execution and to improve system performance. Such is accomplished by replicating remote objects to local systems which access the objects.

With respect to independent claim 1, Appellants maintain that Rich does not teach or suggest at least, "assigning to a father object in a process, for each son object: .... information referring back to said central directory if the son object is not contained in the same process," as recited in claim 1. With respect to this feature, the Examiner alleges in the paragraph bridging pages 5 and 6 of the Office Action dated May 10, 2004:

Rich teaches every node of the tree structure, i.e., including a root
of the tree, using persistent object storage to store information of
the objects, col. 3, lines 13-37. Rich also teaches usage of a
directory/path mechanism, i.e., central directory, to retain
information on the objects, for example, top-level transaction
information, col. 9, lines 3-67. Rich also teaches a node for
multiple transactions in which a node can be a parent for some
transactions while being a child for other transactions. The
parent/child node uses of directory/path mechanism to retain
information on the objects and to refer the directory/path to support
the transaction. The verification of whether a child/parent node in
the same process is performed by the code executing on the server
and client nodes. Col. 9, lines 3-67. Therefore, Rich meets the
claim limitation of "assigning to a father object in a process, for
each son object, information referring back to said central directory
if the son object is not contained in the same process". It is noted
that the claimed subject matter only discloses a central directory,
which can be any directory/path dealing with some other objects
and having no relation to the Applicants' intended implementation
of a tree for distributed objects within several processes. The
different processes can be any processes or a single process and
not limited to several distributed processes. The distributed objects
may not need to be distributed at all. The father object can be a
son of the son object or the son object itself. The assignment of
the father object in a process can be any process and not limited to
the distributed processes. Therefore, [the] Examiner believes the
reference teaches the limitation as disclosed above.

In response, Appellants submit that the Examiner only generally describes the alleged

teachings of Rich, but none of the teachings of Rich, as set forth above, satisfies the specific

limitations of claim 1. Even assuming, *arguendo*, that Rich teaches a node for multiple

transactions in which a node can be a parent for some transactions while being a child for other

transactions, as the Examiner asserts, the specific step of "assigning to a father object in a

process, for each son object, ... information referring back to said central directory if the son is

not contained in the same process," is not satisfied. The Examiner's assertion is based on only

his general alleged description of Rich. Moreover, the Examiner's general statement that a

parent/child node uses a directory/path mechanism to retain information on objects is not

evidence of any prior art teaching, and does not render the above claim limitations satisfied by

Rich either. Furthermore, the Examiner alleges that "verification of whether a child/parent node

in the same process is performed by the code executing on the surveying client nodes," is

discussed at column 9, lines 3-67. Appellants do not quite understand how this statement is

relevant to the above quoted limitation, as the Examiner has not identified which components of

Rich correspond to the claimed father object, the claimed son object, and the claimed same

process. Nowhere has the Examiner identified aspects of Rich that correspond to the specific

claimed components and/or features of the present invention, as set forth in claim 1 (in fact, Rich

contains no such teaching or suggestion). Therefore, at least based on the foregoing, Appellants

submit that the Examiner has not established that each and every limitation, as set forth in claim

1, is satisfied by Rich.

Yet further, the Examiner alleges on the continuation sheet attached to the Advisory

Action dated October 22, 2004:

> Rich clearly discloses claimed components, i.e. father object (e.g.,
> parent node of the tree structure having further child nodes, col. 9,
> lines 3-67), son object (e.g., child node of the parent node of the
> tree structure, col. 9, lines 3-67), same process (e.g., the same code
> executing on the server and client nodes, col. 9, lines 3-67), central
> directory (e.g., use of directory/path information accessed by all
> the nodes for maintaining the tree structure, col. 3, lines 13-37).

In response, Appellants maintain that the Examiner's explanation is confusing and still

does not satisfy the particular limitations of claim 1. That is, the Examiner alleges that the parent

nodes of the tree structure in Rich correspond to the claimed father object, and that the child

node of the parent node of the tree structure of Rich corresponds to the claimed son object.

However, claim 1 recites that information corresponding to a physical address is assigned to a

father object in a process, for each son object, if the son object is contained in a same process as

the father object. Therefore, it is evident that the present invention, as recited in claim 1, describes that a father object is in a process and that a certain operation takes place if a son object is contained in the same process as the father object. Rich, on the other hand, does not show the node 401, which allegedly corresponds to the claimed father object, in a process, and does not show nodes 402 and 405, which are child nodes, which allegedly correspond to the claimed son object, contained in a process. Therefore, Rich does not teach or suggest the above discussed limitations of claim 1. Further, on the continuation sheet, the Examiner alleges that the code executing on the nodes, which the Examiner believes corresponds to the father and/or son objects, corresponds to the claimed same process. That is, the Examiner argues that the code is in or is executed in the nodes. However, the Examiner has not demonstrated that the features of claim 1 are satisfied by Rich because claim 1 requires that a father object is in a process (and not the other way around). Therefore, at least based on the foregoing, Appellants maintain that the present invention, as recited in claim 1, is patentably distinguishable over Rich.

With respect to dependent claims 2-8, Appellants submit that these claims are patentable at least by virtue of their respective indirect or direct dependencies from independent claim 1.

    A.2. <u>Rich does not teach or suggest at least, "if said logical name is not in the central directory, the central directory searches its data structure for a logical name with the longest character string equal to a first part of the character string of the logical name received, in order to send to a father object, the request relating to the first object," as recited in claim 2.</u>

Further, Rich does not teach or suggest the requirement of dependent claim 2 that "if said logical name is not in the central directory, the central directory searches its data structure for a logical name with the longest character string equal to a first part of the character string of the logical name received, in order to send to a father object, the request relating to the first object". As similarly argued above, Appellants submit that the Examiner simply generally describes the

alleged teachings of Rich without providing any actual evidence that Rich constitutes a valid

prior art teaching of the explicit claim requirements. The Examiner does not discuss (and Rich

does not teach or suggest) the central directory searching its data structure for a logical name

with the longest character string equal to a first part of the character string of the logical name

received. Further, the Examiner alleges, generally, that a physical node may have nodes of

transaction trees for multiple transactions, and may be a parent for some transactions while being

a child for other transactions. However, this particular allegation, even if, assuming *arguendo*, it

is true, does not satisfy the particular portion of the above quoted limitation that recites "to send

to a father object, the request relating to the first object." Moreover, Appellants again point out

that the Examiner has not established which components or aspects of Rich allegedly correspond

to the claimed father object and/or first object, as set forth in claim 2.

> A.3. <u>Rich does not teach or suggest at least, "the central directory manages the redundancy of the processes by selecting one of several processes containing the requested object," as recited in claim 4.</u>

Further, with respect to claim 4, Appellants submit that Rich does not teach or suggest at

least "the central directory manages the redundancy of the processes by selecting one of several

processes containing the requested object," as recited in claim 4. In response to this particular

argument, the Examiner makes the allegations set forth in the full paragraph on page 8 of the

Office Action dated May 10, 2004. In that paragraph, the Examiner essentially refers to previous

arguments made earlier in the Office Action and cites and recites the language that is recited at

column 13, line 29 through column 14, line 19 of Rich. In response, Appellants submit, as set

forth above, that still nowhere has the Examiner identified corresponding aspects of Rick that

allegedly satisfy the specific claim components and/or features of the claimed invention.

Further, the cited portion of Rich does not even relate to or involve managing the redundancy of

processes, as recited in claim 4. Therefore, at least for these reasons, Appellants submit that

Rich does not satisfy the limitations of claim 4.

At least based on the foregoing, Applicant submits that claims 1-8 are patentably

distinguishable over Rich.

Appellants submit that claim 9 is patentable at least by virtue of its dependency from

independent claim 1. The AAPA does not make up for the deficiencies of Rich. Even taken as a

whole, the combined teachings of Rich and the APA fail to meet all the requirements of claim 1.
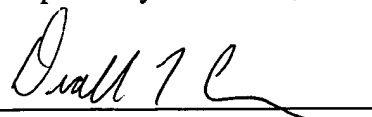
A.4. Conclusion

The Examiner's burden is to make out at least a *prima facie* case, based on evidence of

teachings in the prior art. This the Examiner has not done and, in a rejection over Rich, cannot

do.

Appellants submit that, at least based on the foregoing, the present invention, as recited in

each of claims 1-9, is patentably distinguishable over Rich.

Unless a check is submitted herewith for the fee required under 37 C.F.R. §41.37(a) and

1.17(c), please charge said fee to Deposit Account No. 19-4880.

The USPTO is directed and authorized to charge all required fees, except for the Issue

Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any

overpayments to said Deposit Account.

Respectfully submitted,

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Diallo T. Crenshaw
Registration No. 52,778

Date: January 3, 2005

13

CLAIMS APPENDIX

CLAIMS 1-9 ON APPEAL:

1. A method of implementing a tree of distributed objects in different processes, wherein a central directory is adapted to store information on objects in a data structure at the root of the tree, said method comprising assigning to a father object in a process, for each son object:

information corresponding to a physical address if a son object is contained in same process, or

information referring back to said central directory if the son object is not contained in the same process.

2. The method according to claim 1, wherein if the central directory receives a request for access to a first object identified by a logical name identifying a logical access path of said first object from the central directory, the central directory searches its data structure for the logical name received in order to send the request directly to said first object or, if said logical name is not in the central directory, the central directory searches for a logical name with the longest character string equal to a first part of the character string of the logical name received, in order to send to a father object, the request relating to the first object, by providing said father object with information corresponding to the logical access path of the first object relative to the father object.

3. The method according to claim 2, wherein the father object which receives said request sends the request to said first object if it is a son object of its process or returns a message to the central directory.

4. The method according to claim 1, wherein the central directory manages the redundancy of the processes by selecting one of several processes containing the requested object.

5. The method according to claim 1, wherein if a father object of a process receives a request relating to a son object directly it returns that request to the central directory if said son object is not contained in its process.

6. The method according to claim 5, wherein the son object is identified in said request by a logical name defining the logical access path of that object from said father object, and wherein said father object returns said request to the central directory with the character string of said logical name preceded by the character string corresponding to its own logical name defining its logical access path from the central directory.

7. The method according to claim 1, wherein the central directory contains at least information relating to each root object of each process.

8. The method according to claim 1, wherein the method applies to a distributed object environment based on a manager of the CORBA type.

9. The method according to claim 1, wherein the method applies to a distributed object environment based on a manager of the DCOM type.